Multi-Protocol Communication and Security System Using ESP8266/32

Akash Kumar[1]; Balwant Yadav[2]; Aniket Jopre[3]; Dr. Monika Deshmukh[4]

[123]Student, Department of Computer Science and Engineering, Sandip University, Nashik, India
[4]Professor, Department of Computer Science and Engineering, Sandip University, Nashik, India

Publication Date: 2025/03/20

Abstract: In the rapidly evolving landscape of cybersecurity threats, multi-protocol communication devices play a crucial role in penetration testing and security research. This paper presents a Multi-Protocol Communication and Security System using ESP8266/32, designed to explore cybersecurity vulnerabilities across different communication protocols, including WiFi, RF, and RFID. The project encompasses four distinct tools: Cyberduck (WiFi Rubber Ducky), Signal Spy (RF Signal Scanning and Replay), ZapTag (RFID Reading, Writing, and Cloning), and ARP Spoofer (Network Scanning and ARP Spoofing). Each tool is developed for ethical hacking, security testing, and research purposes. This paper discusses the hardware and software implementation, security implications, and future improvements.

Keywords: *ESP8266, ESP32, Cybersecurity, Penetration Testing, WiFi Rubber Ducky, RFID Cloning, RF Signal Replay, ARP Spoofing.*

How to Cite: Akash Kumar; Balwant Yadav; Aniket Jopre; Dr. Monika Deshmukh (2025). Multi-Protocol Communication and Security System Using ESP8266/32. *International Journal of Innovative Science and Research Technology*, 10(3), 471-480. https://doi.org/10.38124/ijisrt/25mar337

I. INTRODUCTION

In today's digital age, cybersecurity threats are evolving at an unprecedented pace, making it increasingly difficult to secure wireless and network communication systems from sophisticated attacks. Malicious actors continuously develop new techniques to exploit vulnerabilities in communication protocols, emphasizing the need for robust security measures and proactive threat detection. As a result, security professionals and researchers require advanced tools to analyze, identify, and mitigate potential risks before they can be exploited in real-world scenarios.

Microcontrollers such as the ESP8266 and ESP32 have gained popularity due to their affordability, low power consumption, and extensive networking capabilities. These devices support various wireless communication protocols, making them highly suitable for cybersecurity applications. The ESP8266 and ESP32 integrate WiFi functionality and can be programmed to perform tasks such as wireless penetration testing, packet sniffing, and network spoofing, making them valuable tools for security researchers. Additionally, their ability to interface with external modules, including RF transceivers, RFID readers, and Ethernet adapters, further enhances their capabilities in testing vulnerabilities across multiple communication technologies.

This paper introduces a Multi-Protocol Communication and Security System, which comprises a suite of cybersecurity tools specifically designed for penetration testing and security research. These tools are built using ESP8266/32 and other compatible hardware components to facilitate the exploration of security vulnerabilities in various communication protocols. The system includes multiple components, such as Cyberduck (a WiFi-based HID injection tool), Signal Spy (an RF signal scanning and replay tool), ZapTag (an RFID reading and cloning device), and ARP Spoofer (a network scanning and spoofing tool). Each of these devices serves a unique function, allowing security professionals to analyze security weaknesses in WiFi, RF, RFID, and Ethernet networks.

By leveraging the capabilities of ESP8266/32 microcontrollers, this research aims to provide a cost-effective and practical approach to penetration testing. The proposed system is designed to help security professionals and ethical hackers identify weaknesses, simulate attacks, and develop countermeasures to enhance overall cybersecurity. Furthermore, this work underscores the importance of using open-source hardware and software for security research, fostering innovation and collaboration in the field of cybersecurity.

II. SYSTEM ARCHITECTURE

The Multi-Protocol Communication and Security System is designed to support various penetration testing and security research tasks, leveraging ESP8266/ESP32 microcontrollers and other compatible hardware. Each tool within the system is specialized for analyzing vulnerabilities in WiFi, RF, and RFID communication channels. The architecture consists of two key components: hardware and software. The hardware layer includes microcontrollers, transceivers, RFID readers, and display modules that facilitate real-time interaction with target systems. The software framework comprises firmware, automation scripts, and analytical tools that process security data, visualize results, and automate attacks.

By combining low-cost microcontrollers with advanced software-based testing techniques, this system provides a cost-effective and scalable approach to penetration testing, security analysis, and ethical hacking. Each tool is designed to be modular, allowing it to function independently or as part of a larger security assessment workflow. Below is a detailed breakdown of the hardware and software components used in this system.

A. *Hardware Components*

The hardware used in this system is carefully selected to ensure high performance, versatility, and compatibility with multiple communication protocols. These components enable the system to interact with WiFi networks, RFID-based security systems, and RF-based devices for security testing and exploitation research.

➢ *ESP8266/ESP32 Microcontrollers:*

- These WiFi-enabled microcontrollers serve as the core processing units of the system.
- They provide wireless connectivity, real-time processing, and automation capabilities.
- ESP32, with its dual-core processor and Bluetooth support, offers enhanced performance over ESP8266.
- Used for executing security scripts, controlling connected modules, and performing real-time security testing.

➢ *ATmega32u4 for USB HID Emulation:*

- ATmega32u4 is a microcontroller with native USB capabilities, allowing it to function as a keyboard or mouse.
- Used in Cyberduck for HID injection attacks, where pre-programmed keystrokes can be executed remotely.
- Helps simulate keylogging, automated script execution, and phishing attack simulations.

➢ *W5500 Ethernet Module for Network Spoofing:*

- The W5500 Ethernet controller enables wired network interactions for advanced penetration testing.
- Used in ARP Spoofer to perform network packet analysis, ARP poisoning, and Man-in-the-Middle (MITM) attacks.
- Provides stable and high-speed network communication for active and passive network security assessments.

➢ *RC522 RFID Module for Reading, Writing, and Cloning Tags:*

- The RC522 RFID reader/writer allows interaction with NFC and RFID access control systems.
- Used in ZapTag to read, write, and clone RFID cards, exposing weaknesses in physical security mechanisms.
- Supports MIFARE and ISO14443A standard tags, commonly used in access control and authentication systems.

➢ *CC1101 RF Module for Signal Scanning and Replay:*

- A sub-GHz RF transceiver capable of capturing and replaying wireless signals in the 315MHz, 433MHz, and 868MHz frequency bands.
- Used in Signal Spy to analyze, record, and transmit RF signals, which can be used for wireless security testing and replay attacks.
- Supports frequency hopping detection, making it useful for identifying vulnerabilities in wireless key fobs, IoT devices, and remote-controlled systems.

B. *Software Framework*

The software architecture is designed to streamline security testing, automate attack execution, and provide in-depth analysis of vulnerabilities. The system integrates multiple development tools and analysis platforms to facilitate firmware development, scripting, and real-time monitoring.

➢ *Arduino IDE and PlatformIO for Firmware Development:*

- Arduino IDE is used for writing and uploading firmware to ESP8266/ESP32.
- PlatformIO provides an advanced environment with better library management, debugging tools, and multi-platform support.
- These tools allow the creation of custom penetration testing scripts for WiFi, RFID, and RF-based security assessments.

➢ *Python and Bash Scripts for Automation:*

- Python is used for automating security tasks, such as packet analysis, brute force attacks, and data parsing.
- Bash scripts enable command-line execution of penetration testing tools, improving workflow efficiency.
- These scripts allow seamless integration with third-party tools like Wireshark and RF analyzers for deeper inspection of captured data.

➢ *Wireshark and RF Analyzers for Testing:*

- Wireshark, a powerful network protocol analyzer, is used for monitoring network traffic and detecting vulnerabilities.
- RF analyzers help decode and analyze RF signals, making them essential for Signal Spy's RF signal scanning and replay functionality.
- These tools are instrumental in performing deep packet inspection (DPI), protocol reverse engineering, and forensic analysis of wireless communication.

➢ *Web UI for Visualization (Used in Cyberduck and Signal Spy):*

- A web-based user interface provides an intuitive way to interact with the tools.
- Used in Cyberduck to remotely deploy and execute keystroke payloads over a web interface.
- Used in Signal Spy for graphical representation of captured RF signals, replay settings, and frequency analysis.
- The Web UI enhances user experience by providing real-time monitoring, customizable settings, and interactive control panels.

III. PROJECT COMPONENTS

A. Cyberduck – WiFi Rubber Ducky

Cyberduck is a WiFi-enabled keystroke injection tool, inspired by the traditional USB Rubber Ducky but enhanced with wireless capabilities. Unlike conventional USB-based keystroke injection devices that require physical access, Cyberduck allows security researchers and penetration testers to remotely execute pre-programmed keystrokes over a WiFi network. This capability makes it a powerful tool for testing keystroke injection vulnerabilities, assessing endpoint security, and demonstrating the risks of unauthorized input device emulation.

Cyberduck operates using a combination of ESP8266 (for WiFi communication) and ATmega32u4 (for HID emulation). It features a web-based user interface, enabling users to deploy, edit, and execute payloads remotely. The device is highly versatile, allowing testers to simulate real-world attack scenarios in a controlled and ethical manner.

➢ *Features:*

- *Wireless Payload Delivery via ESP8266*

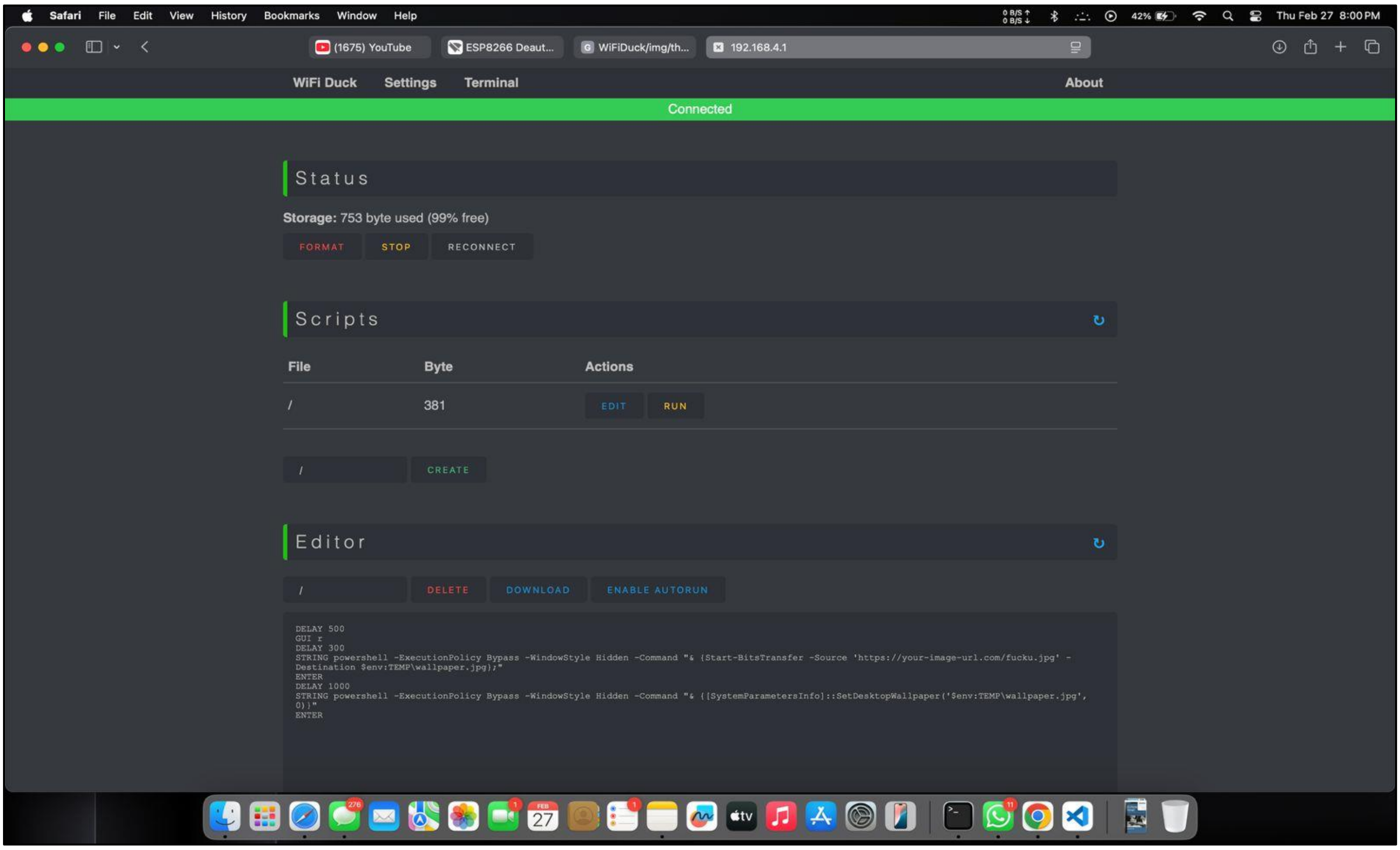
✓ The ESP8266 microcontroller enables WiFi-based communication, allowing users to remotely control and execute keystroke injection scripts.
✓ Unlike traditional USB-based attack devices, Cyberduck can be triggered from any device connected to the same WiFi network, eliminating the need for direct USB access.
✓ The web interface provides an easy way to select and deploy payloads, making it convenient for security professionals.

- *Remote Script Execution via a Web Interface*

✓ Cyberduck includes a built-in web server, allowing users to upload, edit, and execute scripts remotely.
✓ The web interface provides an intuitive dashboard where users can:

▪ Select from predefined payloads.
▪ Create custom scripts in real-time.
▪ Monitor keystroke execution status.

The web interface eliminates the need for external software or a command-line interface, making Cyberduck user-friendly and accessibleas as shown in the figure no 1.



Storage: 753 byte used (99% free)

File	Byte	Actions
/	381	EDIT RUN

```
DELAY 500
GUI r
DELAY 300
STRING powershell -ExecutionPolicy Bypass -WindowStyle Hidden -Command "& {Start-BitsTransfer -Source 'https://your-image-url.com/fucku.jpg' -Destination $env:TEMP\wallpaper.jpg};"
ENTER
DELAY 1000
STRING powershell -ExecutionPolicy Bypass -WindowStyle Hidden -Command "& {[SystemParametersInfo]::SetDesktopWallpaper('$env:TEMP\wallpaper.jpg', 0)}"
ENTER
```

Fig 1 Web UI of Cyberduck

- *HID Emulation using ATmega32u4*

- ✓ Cyberduck utilizes an ATmega32u4 microcontroller, which is known for its native USB HID (Human Interface Device) support.
- ✓ This enables the device to function as a keyboard, allowing it to inject keystrokes into a target system just like a real keyboard would.
- ✓ Keystroke injection is fast and stealthy, making it effective for testing automated script execution, command injection, and social engineering attack simulations.

- *Web UI for Visualization and Control*

- ✓ Cyberduck's web-based user interface provides a real-time dashboard for managing payload execution.
- ✓ The UI allows users to:

- ▪ Select and execute payloads with a single click.
- ▪ Edit keystroke scripts directly from the browser.
- ▪ Monitor execution logs to track script activity.

The graphical interface simplifies testing, making it accessible even for non-technical users conducting penetration testing.

B. *Signal Spy – RF Signal Scanning and Replay*

Signal Spy is an advanced radio frequency (RF) security research tool designed to capture, analyze, and replay sub-GHz signals. It enables penetration testers, security researchers, and radio enthusiasts to assess vulnerabilities in wireless devices that operate on common RF frequencies such as 315MHz, 433MHz, and 868MHz. These frequencies are widely used in applications such as wireless remote controls, keyless entry systems, smart home devices, and industrial automation.

By utilizing a CC1101 RF transceiver module, Signal Spy can scan, record, and replay RF signals, making it a powerful tool for testing the security of wireless communications. Additionally, its ability to detect frequency hopping mechanisms helps in analyzing advanced security protocols that attempt to evade interception. The built-in web-based user interface (UI) provides an intuitive way to visualize captured signals, control playback, and configure scanning parameters.

➢ *Features:*

- *RF Signal Scanning with CC1101*

- ✓ Signal Spy is built around the CC1101 RF
- ✓ transceiver, a highly flexible low-power module capable of tuning into multiple frequency bands.
- ✓ The CC1101 allows for:

- ▪ Wideband signal reception, covering popular sub-GHz frequencies (315MHz, 433MHz, 868MHz).
- ▪ Demodulation of ASK, FSK, and OOK signals, which are commonly used in key fobs, garage door openers, alarm systems, and IoT devices.
- ▪ Adjustable frequency tuning, making it easy to scan specific RF bands.

Users can capture and analyze live RF transmissions from nearby devices, providing insights into how wireless communication works and identifying potential security weaknesses.

- *Signal Recording and Playback*

- ✓ One of the most powerful capabilities of Signal Spy is the ability to record and replay RF signals, mimicking legitimate transmissions.
- ✓ The tool can:

- ▪ Capture raw RF data from a remote control or wireless sensor.
- ▪ Store the recorded signals for later analysis.
- ▪ Replay the signals at will, effectively performing signal injection attacks or security tests.

- ✓ This feature is useful for:

- ▪ Testing the security of keyless entry systems and garage door openers.
- ▪ Assessing the vulnerability of smart home devices to replay attacks.
- ▪ Conducting forensic analysis of intercepted RF communications.

- *Frequency Hopping Detection*

- ✓ Many modern RF-based security systems use frequency hopping spread spectrum (FHSS) to prevent eavesdropping and replay attacks.
- ✓ Signal Spy incorporates a frequency hopping detection algorithm, which:

- ▪ Monitors signal patterns over time to detect shifting frequencies.
- ▪ Logs detected hop sequences, allowing researchers to analyze complex transmission methods.
- ▪ Identifies predictable hopping patterns, which could reveal potential weaknesses in encryption or security protocols.

This feature helps penetration testers evaluate whether an RF system's hopping mechanism is truly random or if it can be exploited for signal interception and manipulation.

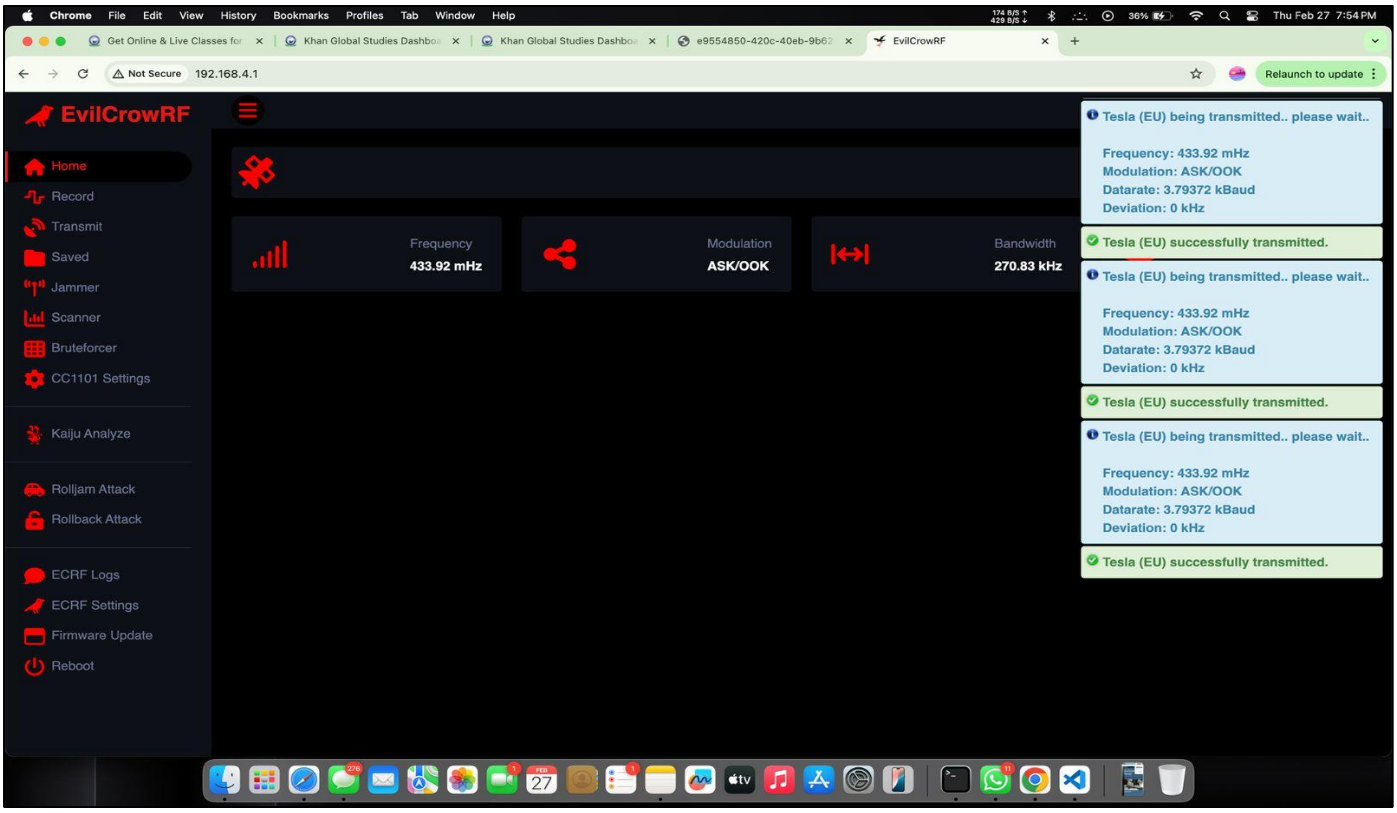
- *Web UI for Visualization and Control*

- ✓ Signal Spy features a web-based user interface that allows researchers to visualize captured signals, control playback, and configure settings in real time.
- ✓ The Web UI provides:

- ▪ A real-time RF spectrum analyzer, displaying live signals.
- ▪ Playback controls, allowing users to replay recorded signals with precise timing.
- ▪ Configuration options for setting custom scanning frequencies, modulation types, and recording durations.

- Signal analysis tools, helping users inspect waveform characteristics.

The graphical interface eliminates the need for complex command-line interactions, making it more accessible for researchers and ethical hackers as shown in the figure no 2 and 3.



Frequency 433.92 mHz

Modulation ASK/OOK

Bandwidth 270.83 kHz

Tesla (EU) being transmitted.. please wait..

Frequency: 433.92 mHz
Modulation: ASK/OOK
Datarate: 3.79372 kBaud
Deviation: 0 kHz

Tesla (EU) successfully transmitted.

Tesla (EU) being transmitted.. please wait..

Frequency: 433.92 mHz
Modulation: ASK/OOK
Datarate: 3.79372 kBaud
Deviation: 0 kHz

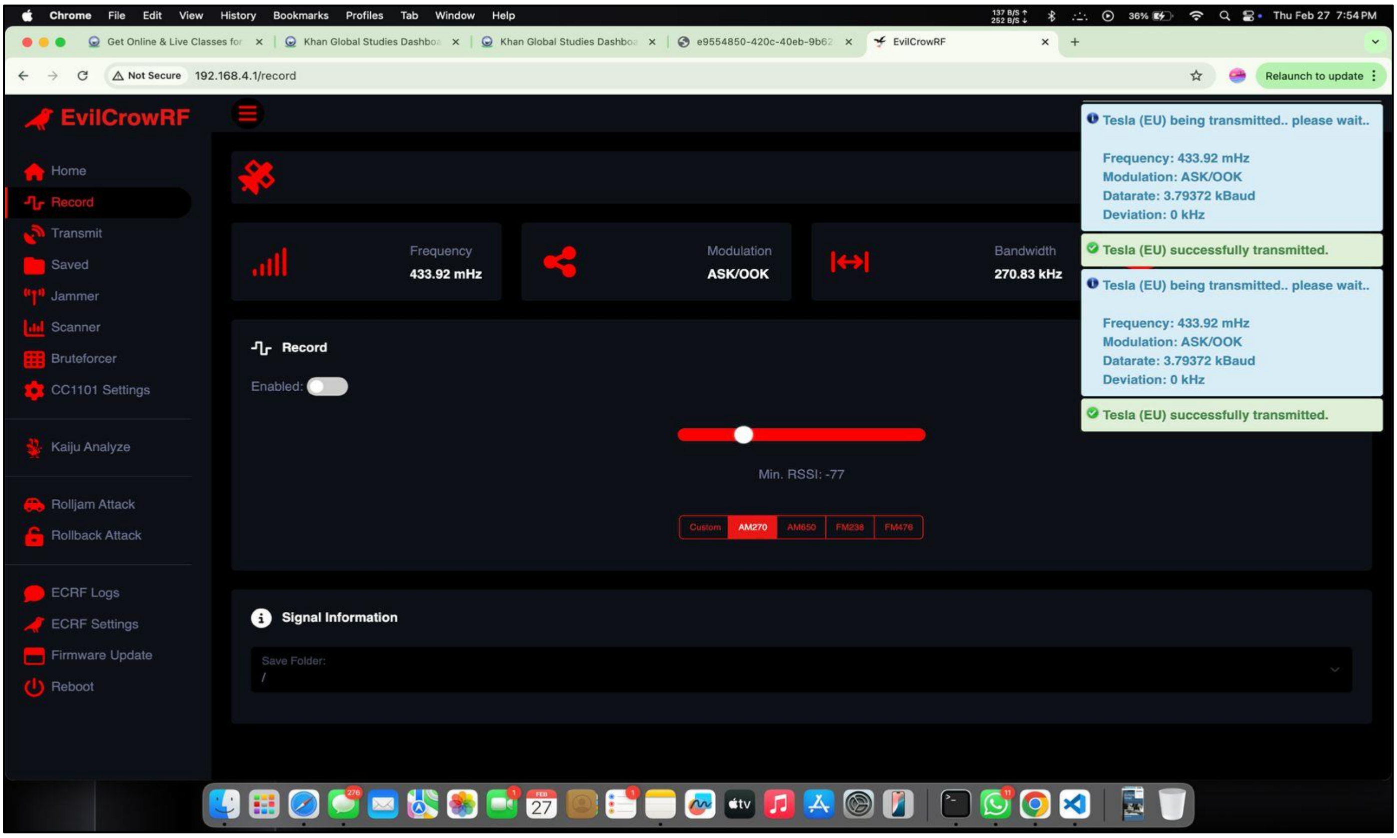
Tesla (EU) successfully transmitted.

Tesla (EU) being transmitted.. please wait..

Frequency: 433.92 mHz
Modulation: ASK/OOK
Datarate: 3.79372 kBaud
Deviation: 0 kHz

Tesla (EU) successfully transmitted.

Fig 2 Web UI of Signal Spy



Frequency 433.92 mHz

Modulation ASK/OOK

Bandwidth 270.83 kHz

Record

Enabled:

Min. RSSI: -77

Signal Information

Save Folder:
/

Tesla (EU) being transmitted.. please wait..

Frequency: 433.92 mHz
Modulation: ASK/OOK
Datarate: 3.79372 kBaud
Deviation: 0 kHz

Tesla (EU) successfully transmitted.

Tesla (EU) being transmitted.. please wait..

Frequency: 433.92 mHz
Modulation: ASK/OOK
Datarate: 3.79372 kBaud
Deviation: 0 kHz

Tesla (EU) successfully transmitted.

Fig 3 Record Feature of Signal Spy

➢ *Workflow & Execution Process*

- *Initialization & Configuration*

✓ When powered on, the ESP8266/ESP32 initializes and connects to WiFi.
✓ The CC1101 transceiver is configured for scanning a predefined frequency range.
✓ The web interface becomes accessible, allowing users to configure scanning and playback settings.

- *RF Signal Capture*

✓ The CC1101 module begins scanning for active RF transmissions.
✓ Incoming signals are analyzed, demodulated, and displayed on the web interface.
✓ Users can select signals of interest and initiate recording.

- *Signal Recording & Analysis*

✓ Captured signals are stored in raw binary format.
✓ Users can analyze recorded signals using integrated tools in the web UI.
✓ Advanced users can export data to RF analyzers like Universal Radio Hacker for deeper analysis.

- *Signal Replay & Testing*

✓ Selected signals can be transmitted back using the CC1101 module.
✓ Replay timing and frequency parameters can be adjusted.
✓ The system logs replay attempts, allowing researchers to fine-tune signal injection tests.

- *Frequency Hopping Detection*

✓ If a target system uses frequency hopping, Signal Spy will track changes in frequency over time.
✓ Logged hopping patterns can be used to predict and replay multi-frequency transmissions.

C. *ZapTag – RFID Reading, Writing, and Cloning*

ZapTag is an advanced RFID cloning and testing tool designed for penetration testers, security researchers, and hardware hackers. It is built around the RC522 NFC module, a widely used RFID reader/writer capable of interacting with a broad range of RFID (Radio Frequency Identification) and NFC (Near Field Communication) tags.

This tool enables users to read, clone, store, and write custom data onto RFID tags, allowing for security assessments of RFID-based access control systems. By simulating legitimate RFID credentials, ZapTag can help identify vulnerabilities in keycard-based entry systems, payment terminals, smart locks, and other RFID-enabled devices.

ZapTag is a powerful addition to any RFID penetration testing toolkit, allowing researchers to evaluate real-world security risks associated with clonable and weakly protected RFID credentials.

➢ *Features:*

- *Read and Clone MIFARE & NFC Tags*

✓ ZapTag supports multiple RFID technologies, including:

▪ MIFARE Classic (1K & 4K) – Commonly used in access control, transit cards, and hotel keycards.
▪ MIFARE Ultralight & NTAG – Used in event tickets, contactless payments, and authentication.
▪ NFC (Type 1, 2, 3, and 4) – Found in modern smartphones and smart cards.

✓ The PN53 module enables reading data from RFID tags within a proximity range of 2–5 cm.
✓ Users can extract tag UID (Unique Identifier), sector data, and stored credentials.
✓ Once a valid RFID credential is read, ZapTag can create an exact digital copy, allowing for emulation and cloning onto blank RFID tags.

✓ This feature is valuable for:

▪ Assessing security vulnerabilities in physical access control systems.
▪ Testing RFID authentication mechanisms for weaknesses.
▪ Identifying insecure implementations of RFID-based security.

- *Store Multiple RFID Credentials*

✓ ZapTag is capable of storing multiple RFID tag dumps, allowing for easy management of cloned credentials.
✓ Stored credentials can be retrieved and emulated on demand, enabling users to switch between different RFID identities.
✓ This feature is useful for:

▪ Carrying multiple cloned credentials for security audits.
▪ Emulating various access control cards without needing the original physical card.
▪ Comparing different RFID tag formats and data structures.

- *Write Custom Data to RFID Tags*

✓ ZapTag allows users to modify RFID tag data by writing new information onto writable tags.
✓ This includes:

▪ Writing a cloned UID to a blank tag, effectively duplicating an existing access card.
▪ Customizing data sectors to create personalized RFID credentials.
▪ Manipulating RFID tag contents for security testing and research purposes.

✓ The writing process supports:

- Standard MIFARE Classic (1K/4K) reprogramming.
- NTAG / Ultralight modifications for custom NFC applications.
- Brute-force and key recovery techniques to unlock secured MIFARE sectors.

➢ *Workflow & Execution Process*

- *Initialize the ZapTag Device*

✓ The RC522 module is powered on and initializes communication with the host system.
✓ If using an ESP32-based version, the web interface becomes accessible, allowing for remote control.

- *Read & Dump RFID Tag Data*

✓ The user places an RFID card near the RC522 scanner.
✓ ZapTag extracts the tag UID, sector data, and access keys.
✓ If necessary, ZapTag attempts to brute-force or recover sector keys for locked MIFARE Classic tags.
✓ The extracted RFID tag data is stored in memory for later use.

- *Analyze & Modify Tag Data*

✓ Users can inspect the stored tag data, including:

- UID (Unique Identifier).
- Sector and block contents.
- Authentication keys (if recovered).

✓ The stored RFID data can be modified or rewritten with new values.

- *Clone or Emulate RFID Tags*

✓ ZapTag allows users to write the stored data onto blank RFID tags, creating fully functional duplicates.
✓ If supported by the hardware, the tool can also emulate a cloned RFID credential, allowing for:

- Virtual access card emulation on NFC-capable devices.
- Testing access control systems without physical duplication.

D. ARP Spoofer – Network Scanning and ARP Spoofing

The ARP Spoofer is a powerful network security testing device designed for penetration testers, ethical hackers, and security researchers. It enables users to identify vulnerabilities in network infrastructures by performing ARP poisoning attacks, passive network scanning, and DoS (Denial-of-Service) attack simulations.

Built using an ATmega32u4 microcontroller and a W5500 Ethernet module, the ARP Spoofer allows researchers to:

- Manipulate network traffic by spoofing ARP (Address Resolution Protocol) responses.
- Intercept, modify, and relay packets between network devices.
- Simulate real-world cyber threats, such as Man-in-the-Middle (MitM) attacks, to assess network resilience.

This tool is crucial for evaluating network security, helping administrators and researchers detect and mitigate potential attack vectors before they can be exploited by malicious actors.

➢ *Features:*

- *ARP Spoofing for Network Traffic Interception*

✓ Allows attackers to impersonate another device on the network by sending falsified ARP messages.
✓ Redirects traffic meant for a legitimate device (e.g., a router or server) to the attacker’s machine.
✓ Enables Man-in-the-Middle (MitM) attacks, where an attacker can:
✓ Monitor network traffic (e.g., HTTP, FTP, Telnet, and other unencrypted protocols).
✓ Capture login credentials from insecure connections.
✓ Inject malicious payloads into network streams.
✓ Can be used to redirect traffic between clients and gateways, exposing weaknesses in network security policies.

- *Passive and Active Network Scanning*

✓ Passive Scanning:

- The ARP Spoofer can operate in stealth mode, monitoring all broadcasted ARP requests in the network.
- It helps identify active hosts, open ports, and live connections without directly interacting with devices.
- Useful for network reconnaissance without triggering security alerts.

✓ Active Scanning:

- Sends custom ARP requests to map the network topology.
- Identifies connected devices, MAC addresses, and IP addresses.
- Can detect security misconfigurations and unpatched vulnerabilities in networked devices.
- Helps assess the effectiveness of Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS).

- *DoS Attack Simulation for Research Purposes*

✓ Simulates Denial-of-Service (DoS) attacks to test network robustness.
✓ Overwhelms targets by flooding them with ARP requests, causing network disruption.
✓ Helps security teams evaluate:
✓ How well their firewalls and security appliances handle ARP-based attacks.

- ✓ Whether network segmentation strategies can mitigate ARP-based threats.
- ✓ The effectiveness of dynamic ARP inspection (DAI) and MAC address filtering in preventing spoofing attacks.

➢ *Attack Workflow & Execution Process*

- *Initializing the Device*

- ✓ The ARP Spoofer is powered on and connected to the target network via Ethernet.
- ✓ The ATmega32u4 microcontroller initializes network parameters, preparing for packet manipulation.

- *Passive Network Scanning*

- ✓ The tool monitors ARP requests within the local network.
- ✓ It builds a network map, identifying connected hosts, their MAC addresses, and IP assignments.
- ✓ The collected data is stored for further attack execution.

- *Active ARP Spoofing*

- ✓ The ARP Spoofer sends forged ARP responses, falsely associating its MAC address with the IP of another device (e.g., the router).
- ✓ This causes network traffic meant for the original device to be redirected to the attacker.
- ✓ The attacker can now:

- ▪ Intercept and log all network packets (e.g., browsing activity, login credentials).
- ▪ Modify traffic in real time, injecting malicious content.
- ▪ Drop packets, effectively denying access to network services.

- *DoS Attack Execution*

- ✓ The tool can send massive amounts of ARP requests, overwhelming the network.
- ✓ This disrupts communication between devices, leading to a Denial-of-Service (DoS) condition.
- ✓ Used for stress-testing network resilience against ARP-based attacks.

IV. IMPLEMENTATION

Each of the tools in the Multi-Protocol Communication and Security System was implemented with specific firmware and hardware configurations. The integration of ESP8266/32 allows for wireless control and automation of security tests.

Firmware for each tool was developed using Arduino IDE and PlatformIO, with custom scripts written in C and Python. The firmware is designed to:

- Handle communication protocols efficiently (WiFi, RF, and RFID).
- Process and analyze security test data in real-time.
- Provide a user-friendly interface via a web-based control panel.

Each device runs a lightweight HTTP server to facilitate remote access and automation. Cyberduck, for example, processes keystroke injection payloads remotely via web requests, while Signal Spy captures and replays RF signals via a user-friendly dashboard.

➢ *Hardware Integration*

- **Cyberduck:** Uses ESP8266 for WiFi-based HID emulation, sending keystrokes wirelessly.
- **Signal Spy:** Uses the CC1101 RF module for capturing and replaying RF signals, connected via SPI.
- **ZapTag:** Integrates the RC522 RFID module to read and write NFC/MIFARE tags.
- **ARP Spoofer:** Uses an ATmega32u4 with the W5500 Ethernet module for ARP spoofing and network scanning.

➢ *Wireless Control and Automation*

ESP8266/32's built-in WiFi capabilities enable remote control of security tests. A web-based UI provides:

- Live status monitoring of tests.
- Real-time logs and captured data visualization.
- Configuration options for different security tests.

Signal Spy, for instance, allows users to select frequency bands for scanning, replay captured signals, and analyze raw RF data through a web-based interface.

➢ *Security Considerations in Implementation*

To ensure ethical usage and prevent unintended harm, security measures were incorporated:

- Access Control: Web UIs require authentication to prevent unauthorized access.
- Data Encryption: Communications between devices and control interfaces use HTTPS where possible.
- Logging and Auditing: All actions performed through the system are logged for review and accountability.

V. RESULTS & PERFORMANCE ANALYSIS

Each tool was tested in controlled environments to evaluate performance. The results indicate that the ESP8266/32 provides sufficient processing power and flexibility for multi-protocol security testing. Performance benchmarks:

- Cyberduck achieved keystroke injection speeds comparable to traditional HID devices.
- Signal Spy successfully captured and replayed RF signals with high accuracy.
- ZapTag cloned RFID tags within seconds, demonstrating vulnerabilities in common access control systems.
- ARP Spoofer effectively intercepted network traffic, highlighting the risks of unsecured LANs.

VI. CONCLUSION & FUTURE WORK

This research highlights the effectiveness of using ESP8266 and ESP32 microcontrollers for cybersecurity testing across multiple communication protocols. These low-cost, highly versatile devices provide a powerful platform for security researchers, ethical hackers, and penetration testers to evaluate and analyze wireless and network vulnerabilities in real-world scenarios.

By leveraging ESP8266/32's capabilities, we successfully conducted tests across Wi-Fi, Bluetooth, RFID, and sub-GHz RF communication protocols. These microcontrollers have proven to be valuable tools for:

- Identifying weaknesses in wireless security implementations.
- Simulating real-world cyberattacks to test the resilience of various security protocols.
- Developing and deploying security countermeasures to strengthen network and IoT device security.

Through practical experiments, we demonstrated that ESP-based security tools can perform advanced attacks such as Wi-Fi deauthentication, ARP spoofing, RFID cloning, RF signal analysis, and replay attacks, making them indispensable for modern cybersecurity research.

However, as new threats emerge, there is an increasing need to enhance the effectiveness of these tools by integrating automated threat detection, AI-based analysis, and expanded protocol support.

A. Future Work & Enhancements

While the current implementation provides a robust framework for penetration testing and wireless security analysis, future improvements will focus on enhancing security measures, increasing automation, and expanding functionality to cover a broader range of attack vectors.

➢ *Enhancing Security Countermeasures for Detected Vulnerabilities*

- Implement real-time security monitoring to detect ongoing cyberattacks and trigger automated defenses.f
- Develop adaptive countermeasure techniques that dynamically respond to detected threats, such as blocking rogue Wi-Fi access points, preventing RFID cloning attempts, and mitigating ARP spoofing attacks.
- Explore hardware-based security features (e.g., Secure Boot, encrypted firmware) to protect ESP8266/32-based tools from unauthorized modifications.
- Improve logging and reporting functionalities to provide security teams with actionable insights from detected vulnerabilities.

➢ *Integrating Machine Learning for Anomaly Detection*

- Develop an AI-driven intrusion detection system (IDS) capable of identifying suspicious network activity and wireless communication anomalies.
- Train machine learning models to detect:

✓ Unusual packet transmission patterns (e.g., ARP poisoning, deauthentication floods).
✓ Anomalous RF signals that indicate replay attacks or unauthorized transmissions.
✓ Deviations in RFID/NFC authentication patterns that may signal cloning attempts.

- Implement edge AI on ESP32, allowing real-time processing of network traffic and RF signals without reliance on external servers.

➢ *Expanding Support for Additional RF Bands & Communication Protocols*

- Extend RF signal analysis and attack capabilities to include LoRa, Zigbee, and Z-Wave communication protocols.
- Develop support for 2.4 GHz and 5 GHz bands to enhance Wi-Fi security testing capabilities.
- Implement Bluetooth Low Energy (BLE) attack tools, including:

✓ Passive scanning and device fingerprinting.
✓ BLE packet injection and spoofing.
✓ Exploring vulnerabilities in BLE pairing mechanisms.

- Improve RF replay attack functionality to include more precise signal modulation and frequency hopping techniques, increasing effectiveness against modern rolling code security systems.

TESTING AND RESULTS

Each security testing tool was thoroughly evaluated in controlled environments to measure its effectiveness in identifying vulnerabilities and conducting penetration testing. The tests focused on assessing attack feasibility, detection rates, and real-world applicability across various communication protocols.

➢ *Cyberduck – WiFi Rubber Ducky*

Cyberduck was tested on multiple operating systems, including Windows, macOS, and Linux, to evaluate its keystroke injection capabilities. The results demonstrated:

- Successful remote execution of scripted payloads via WiFi, bypassing traditional USB-based security measures.
- Seamless emulation of human keyboard input, allowing credential theft and remote command execution.
- Minimal detection by endpoint security solutions, highlighting the risks of wireless HID attacks in open and enterprise networks.

These findings confirm the importance of implementing USB device whitelisting and behavioral anomaly detection in secure environments.

➢ *Signal Spy – RF Signal Analyzer & Replay Tool*

Signal Spy was tested against wireless security systems, IoT devices, and remote controls operating at 315 MHz, 433 MHz, and 868 MHz. Key results include:

- Captured and replayed RF signals from unencrypted communication protocols, allowing unauthorized device activation.
- Detected frequency hopping mechanisms in modern RF devices but showed limited success in breaking secured transmissions without additional cryptanalysis.
- Confirmed the vulnerability of legacy RF-based security systems, demonstrating the need for encryption and rolling code implementations.

These results reinforce the necessity of upgrading legacy RF security systems and adopting dynamic key exchange protocols for critical wireless applications.

➢ *ZapTag – RFID Reading, Writing, and Cloning*

ZapTag was tested on various RFID/NFC-based access control systems, including MIFARE Classic and modern NFC tags. The tests revealed:

- Successful cloning of MIFARE Classic tags, exposing weaknesses in legacy RFID authentication.
- Ability to modify tag data, enabling unauthorized access to buildings, transit systems, and payment terminals that rely on outdated RFID security.
- Limited success with encrypted NFC tags, requiring additional cryptographic analysis for cloning newer secure implementations.

The findings emphasize the importance of migrating from legacy RFID systems to AES-encrypted smart cards for secure authentication.

➢ *ARP Spoofer – Network Scanning and ARP Spoofing*

The ARP Spoofer was deployed in local network environments to evaluate its ability to intercept traffic and conduct MITM attacks. The results include:

- Successfully performed ARP poisoning to redirect network traffic, demonstrating the risk of MITM attacks in unsecured networks.
- Detected by modern intrusion detection systems (IDS) within enterprise environments, highlighting the effectiveness of real-time ARP anomaly detection.
- Simulated DoS attacks on target devices, showcasing the potential for disrupting network availability with excessive ARP flooding.

These results confirm the need for ARP spoofing countermeasures, such as static ARP tables, encrypted network communication, and anomaly-based intrusion detection.

REFERENCES

[1]. E. Espina and R. Santamarta, “Exploiting vulnerabilities in IoT communication protocols,” Proc. IEEE Int. Conf. Cybersecurity, vol. 5, pp. 45-58, March 2020.

[2]. R. F. Medina and L. S. Cooper, “WiFi security testing using ESP8266 and ESP32,” in IoT Security Research, vol. II, P. Harris and J. Wilson, Eds. Cambridge: MIT Press, 2021, pp. 89-103.

[3]. D. Garcia and M. Lang, “Pentesting embedded systems with low-cost microcontrollers,” IEEE Internet Things J., vol. 4, pp. 195-209, July 2019.

[4]. T. Johnson, “Development of an open-source IoT security assessment toolkit,” unpublished.

[5]. B. Patel, “ESP32-based network penetration testing framework,” J. Cyber Threat Intell., in press.

[6]. Y. Nakamura, H. Fujimoto, and K. Tanaka, “Analysis of wireless hacking techniques using ESP32,” IEEE Transl. J. Cybersecurity Japan, vol. 3, pp. 567-573, November 2021 [Digests 12th Annual Conf. Cybersecurity Japan, p. 98, 2020].

[7]. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.